

Struts2 框架核心配置文件的研究与应用

刘艳春,洪晓慧

(1. 渤海大学 大学计算机教研部,辽宁 锦州 121013;
2. 中国联通公司锦州分公司,辽宁 锦州 121000)

摘要: Struts2 是强大的 Java Web 开源框架,处理过程通过配置文件完成。针对 Struts2 框架配置文件内容多、配置复杂、难以掌握的特点,对核心配置文件进行研究。首先,列出了 Struts2 框架的所有配置文件及其之间的相互关系,指出了 struts.xml 和 struts.properties 是核心配置文件;接着,研究 struts.xml 配置文件,包括 struts.xml 常用配置解析、struts.xml 构建步骤及构建方法;最后研究 struts.properties 配置文件,着重研究 struts.properties 中包含主要的 Struts2 属性及配置方法。文中的研究内容,能够使开发者清晰地理解配置文件的写法与调用关系,有利于团队成员并行工作,提高开发效率。

关键词: Struts2 框架;配置文件;struts.xml;struts.properties

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X (2013)02-0077-05

doi:10.3969/j.issn.1673-2013.02.021

Research and Application of Core Configuration Files of Struts2 Framework

LIU Yan-chun, HONG Xiao-hui

(1. Teaching and Research Institute of College Computer, Bohai University, Jinzhou 121013, China;
2. China Unicom Jinzhou Branch, Jinzhou 121000, China)

Abstract: Struts2 is a powerful open source framework of Java Web, the processing procedure of Struts2 framework has been finished by configuration files. For the characteristic of framework configuration files more content, configuration complexity and hard to master, the two core configuration files are researched. At first list all the configuration files of Struts2 and their relationship. Then research struts.xml configuration file, including struts.xml common configuration analysis, construction steps and building method of the struts.xml. Finally, research configuration files of the struts.properties, focus on the main Struts2 attribute and configuration method contained in struts.properties. The research content in this paper can make developers clear understanding the way of writing and calling relationships of configuration files, it makes for team members to work in parallel and improve efficiency of development.

Key words: Struts2 framework; configuration files; struts.xml; struts.properties

0 引言

Struts 是目前最流行的 Web 框架,使用该框架使开发 MVC 程序非常简单容易^[1]。经过多年发展,Struts 已经成为一个高度成熟的框架,在国内使用非常广泛,成为 Java 程序员必备的一项基本技能。但随着交互性更强的 Web 应用程序 Ajax、以组件为中心来开发 Java Web 用户界面 JSF 等技术的产生和发展,最初的 Struts 暴露出很多明显的缺陷,运用起来越来越不方便。WebWork 作为新出现的 MVC 框架,很好地融合了 Ajax、JSF 等技术,但其在页面导航及标记库等方

面存在缺陷,开发效率较低且系统各个组成部分之间的脉络不够清晰。基于此,Struts 框架整合 WebWork 框架,构建全新的 MVC 框架,形成了 Struts2 框架。Struts2 是 Struts1 和 WebWork 的更新产品^[2,3]。

Struts2 采用的是多层体系结构,包括表示层、业务逻辑层、持久层、数据库层。其中表示层提供用户界面的显示与用户进行交互操作,业务逻辑层进行软件核心业务逻辑的处理,持久层进行对象的持久化操作,数据库层用于保存持久化的数据。这样处理的好处是可以降低系统的耦合度,每个层次间都有明确的分工。最初软件框架的层次间内聚度较强,通过编程实现,层次之间的耦合依赖于具体程序,不容易维护。Struts2 中的每个层次可以各为一体,层次之间通过接口的定义和配置实现松散耦合。请求经过一系列过滤器,通过配置文件 struts.xml 和 struts.properties 的配置返回

收稿日期:2012-06-02;修回日期:2012-09-04

基金项目:辽宁省科学事业公益研究基金(2011004001)

作者简介:刘艳春(1977-),男,辽宁铁岭人,讲师,从事计算机应用的教学与科研工作。

结果^[4,5]。

Struts2 框架的处理过程是通过配置文件来完成的^[6]。对于应用 Struts2 框架的开发人员,需要掌握若干个配置文件,并且开发程序的过程,就是不断地写配置文件的过程,作者结合多年软件开发和教学过程中的实际经验,深入研究核心配置文件,旨在使读者清晰地理解配置文件的写法与调用关系,提高开发效率。

1 Struts2 框架配置文件

Struts2 框架有 6 个配置文件,如表 1 所示。各个配置文件的相互关系如图 1 所示^[7]。在这些配置文件中,最重要的是 struts.xml 和 struts.properties。其他配置文件的应用较少,而且配置方法简单。struts.xml 和 struts.properties 配置复杂,而且有很多技巧,这两个配置文件是 Struts2 框架的核心和基础,因此也称为核心配置文件。

表 1 Struts2 框架的所有配置文件

文件名称	用途
web.xml	Web 部署描述,包括所有必需的 Struts2 组件
struts.xml	主要配置,包括 Result/View 类型,Action 映射,拦截器等
struts.properties	Struts2 的属性配置
struts-default.xml	struts.xml 中应该有的默认配置
velocity.properties	velocity 配置
struts-default.vm	velocity 中应该有的默认配置

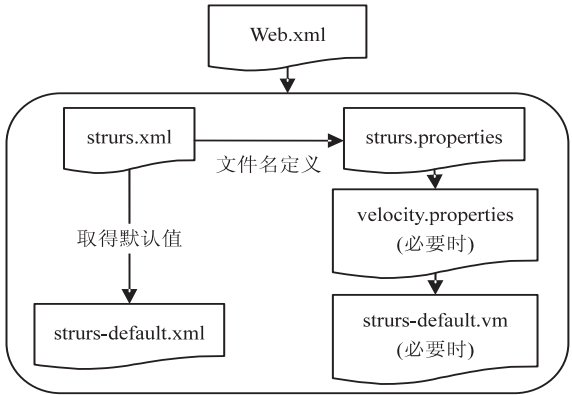


图 1 Struts 框架的配置文件关系图

2 struts.xml 配置文件

Struts2 框架的核心配置文件是 struts.xml,主要负责管理 Struts2 框架的业务控制器 Action 映射以及该 Action 包含的 Result 定义等。struts.xml 这个文件名是默认情况下的文件名,但并不是固定不变的,可以通过 struts.configuration.files 改变为其它的文件名。默认情况下是一个文件,可以使用<include>标签把内容分配到几个文件中,也可以把其他文件的内容导入进来,

被导入的配置文件必须和 struts.xml 具有同样的格式^[8,9]。

2.1 struts.xml 常用配置解析

● 使用<include>标签重用配置文件。

在 Struts2 中提供了一个默认的 struts.xml 文件,但如果当 package、action、interceptors 等配置比较多时,都放到一个 struts.xml 文件不太容易维护^[4]。因此,就需要将 struts.xml 文件分成多个配置文件,然后在 struts.xml 文件中使用<include>标签引用这些配置文件。这样做的优点是:结构更清晰,更容易维护配置信息;配置文件可以复用。如果在多个 Web 程序中都使用类似或相同的配置文件,那么可以使用<include>标签来引用这些配置文件,这样可以减少工作量。

● action 的别名。

在默认情况下,Struts2 会调用动作类的 execute 方法。但有些时候,需要在一个动作类中处理不同的动作。也就是用户请求不同的动作时,执行动作类中的不同的方法。为了达到这个目的,可以在<action>标签中通过 method 方法指定要指行的动作类的方法名,并且需要为不同的动作起不同的名子(也称为别名)。

● 为 action 指定参数。

在 Struts2 中还可以为 action 指定一个或多个参数。在 struts1.xml 中可以使用<action>标签的 parameter 属性为其指定一个 action 参数,如果要指定多个,就只能通过逗号(,)或其他分隔符将不同的参数隔开。而在 Struts2 中可以通过<param>标签指定任意多个参数。

● 选择 result 类型。

在默认时,<result>标签的 type 属性值是“dispatcher”(实际上就是转发,forward)。开发人员可以根据自己的需要指定不同的类型,如 redirect、stream 等。如下面代码所示:

```
<result name="save" type="redirect">
    /result.jsp
</result>
```

result-type 可以在 struts2-core-2.0.11.1.jar 包或 Struts2 源代码中的 struts-default.xml 文件中找到,在这个文件中找到<result-types>标签,所有的 result-type 都在里面定义了^[10]。

● 全局 result。

有些时候一个<result>被很多<action>使用,这时可以使用<global-results>标签来定义全局的<result>。

如果<action>中没有相应的<result>,Struts2 就会使用全局的<result>。

2.2 struts.xml 构建步骤

struts.xml 文件构建步骤如图 2 所示。

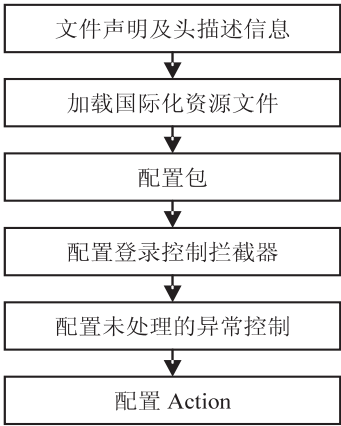


图2 struts.xml 文件构建步骤

● 步骤 1:文件声明及头描述信息。

文件声明及头描述信息如下:

```
<!-- DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD
    Struts Configuration 2.0//EN"
    "file:///D:/JavaPlatform/Lib/struts-2.0.dtd">
```

● 步骤 2:加载国际化资源文件。

可以在 struts.properties 中配置,也可以在 struts.xml 中配置。在 struts.xml 中配置是加入一个常量,指定国际化资源文件的 baseName 为 messageResource。如下:

```
<constant name="struts.custom.i18n.resources"
    value="messageResource"/>
```

● 步骤 3:配置包。

可以定义 Struts2 的默认包空间,如下:

```
<package name="struts-default">
```

默认包空间里定义了 Struts2 内建的 Result 类型,还定义了 Struts2 内建的拦截器,以及由不同拦截器组成的拦截器栈。

也可定义包空间,如下:

```
<package name="ren" extends="struts-default">
```

name 为包空间,可以随意起名,表明包中内容所占用的空间。extends 通常继承 struts-default 的包空间, struts-default.xml 文件在 struts2-core-2.0.8.jar 包里。如果用 struts2 下的 AJAX 插件,用来进行数据校验,则继承 json-default。json-default.xml 文件在 json-plugin-0.7.jar 包里。如下:

```
<package name="ren" extends="json-default">
```

● 步骤 4:配置登录控制拦截器。

拦截器体系是 Struts2 的重要组成部分,完成了框架的大部分操作。拦截器是对调用方法的改进,通过拦截器机制,不仅允许改变被调用的方法体,甚至可以完全改变调用的目标方法。当需要使用某个拦截器

时,只需要在配置文件中配置即可。如果用户的 Action 需要使用某些特定的通用功能,只需要在 struts.xml 中指定拦截器,就可以使拦截器方法在目标方法执行前或者执行后自动执行,从而完成通过操作的动态插入^[2]。

如果在每个 Action 里都有相同的登录控制代码,不利于代码重复利用。采用 Struts2 的拦截器机制进行登录控制。拦截所有用户请求,然后检查用户 Session,如果用户 Session 中包含了用户 ID,即表明用户已经登录;否则系统将自动转入登录页面。这样能保证所有功能只有登录用户才可以使用。

拦截器的配置示例如下:

```
<!--配置登录控制拦截器-->
<interceptors>
<!--定义系统的登录控制的拦截器-->
<interceptor name="authority"
class="org.yeeeku.action.interceptor.AuthorityInterceptor"/>
<!--定义系统 auctionStack 拦截器-->
<interceptor-stack name="auctionStack">
<!--引用系统的默认拦截器栈-->
<interceptor-ref name="defaultStack"/>
<!--引用登录控制拦截器-->
<interceptor-ref name="authority"/>
</interceptor-stack>
</interceptors>
```

● 步骤 5:配置未处理的异常控制。

在 Action 的 execute 方法中使用 try...catch 块来捕捉异常,当捕捉到指定的异常时,系统返回到指定的逻辑视图名,这种方式完全是手动处理异常,非常繁琐,而且可维护性差。

Struts2 的异常处理机制是通过在 struts.xml 文件中配置异常处理,通过配置<exception-mapping.../>元素完成,配置该元素时,需要指定两个属性:

exception:指定异常映射所设置的异常类型;

result:指定 Action 出现异常时,系统转入 result 属性所指向的结果。

例如,当 Struts2 系统捕捉到未处理的异常时,系统自动转入指定的 JSP 页面,并在 JSP 页面中输出异常提示信息,通过这种方法极好地简化了系统的异常控制。配置如下:

```
<!--配置全局的 Result-->
<global-results>
<!--下面定义的结果对所有的 Action 都有效-->
<result name="exception">/error.jsp</result>
<result name="success">/success.jsp</result>
</global-results>
<!--配置全局的异常映射-->
<global-exception-mappings>
```

```
<! --指 Action 抛出 Exception 异常时,转入名为 excep-
tion 的结果-->
<exception-mapping exception = " java. lang. Exception" re-
sult = " exception" />
</global-exception-mappings>
```

如果不需要向用户精确显示每种异常,可在业务方法捕捉全部异常,然后抛出业务异常。

● 步骤 6:配置 Action。

包括处理用户登录的 Action 和正常的 Action。

Action 处理完用户请求后,将返回一个普通字符串,整个普通字符串就是一个逻辑视图名。Struts2 通过配置逻辑视图名和物理视图之间的映射关系,一旦系统收到 Action 返回某个逻辑视图名,系统就会把对应的物理视图呈现给浏览者^[2]。

Action 的 name 属性,既是 Action 的名字,也是该 Action 需要处理的 URL 的前半部分。

Action 的 class 属性,指定了该 Action 的实现类。

Result 的 name 属性,为 Action 处理用户请求结束后,返回的逻辑视图名(普通字符串)。

Result 的 type 属性,为 Struts2 支持的处理结果类型。

处理用户登录 Action 的配置方法如下:

```
<! --配置处理用户登录的 Action -->
<action name = " processLogin"
    class = " org. yeeku. action. LoginAction" >
    <! --配置处理用户登录后的 Result -->
    <result name = " input" >/WEB-INF/jsp/login. jsp</result>
    <result name = " success" >/WEB-INF/jsp/main. jsp</result>
    <result name = " failure" >/WEB-INF/jsp/login. jsp</result>
    <! --显示引用系统默认拦截栈 -->
    <interceptor-ref name = " defaultStack" />
</action>
```

3 struts. properties 配置文件

struts. properties 文件是标准配置文件,用于配置 Struts2 的属性。struts. properties 文件包含了系列的 key-value 对象,是一个标准的 Properties 文件,每个 key 就是一个 Struts2 属性,key 对应的 value 就是一个 Struts2 属性值^[11]。struts. properties 是全局配置文件,但并不是必须的,可以在 struts. xml 中通过<constant>配置这些静态属性,与在 struts. properties 中配置的效果一样,但会使 struts. xml 文件内容过多,不易维护。

3.1 直接配置属性示例

直接配置就是在 struts. properties 文件中配置,配置方法如下:

```
<! --指定加载 Struts2 的配置文件管理器 -->
<! --也可以实现一个 Configuration 接口的类 -->
<! --自己加载 Struts2 配置文件 -->
struts. configuration = org. apache. Struts2. config. DefaultCon-
figuration

<! --设置 Struts2 支持动态方法调用 -->
struts. enable. DynamicMethodInvocation = true
<! --每次 HTTP 请求到达时,系统都重新加载资源文件 -->
->
struts. i18n. reload = true
<! --指定 Web 应用的默认 Locale -->
<! --是国际化与本土化过程中的一个非常重要的概念 -->
->
struts. locale = zh_CN

<! --指定 Web 应用的默认编码集 -->
<! --当获取中文请求参数值,设置为 GBK 或者 GB2312 -->
->
struts. i18n. encoding = GBK
<! --指定 Spring 框架的自动装模式 -->
<! --默认根据 Bean 的 name 属性自动装配 -->
struts. objectFactory. spring. autoWire = name
<! --设置 Struts 2 允许在 Action 名中使用斜线 -->
struts. enable. SlashesInActionNames = true

<! --该属性设置浏览器是否缓存静态内容 -->
<! --当应用处于开发阶段时设置为 true,否则设计为
false -->
struts. serve. static. browserCache = false
<! --指定 Struts2 框架默认加载的配置文件 -->
<! --多个配置文件之间以逗号(,) 隔开 -->
struts. configuration. files = struts-student. xml, struts-teachet.
xml

<! --指定将 HTTP 请求映射到指定 Action 的映射器 -->
struts. mapper. class = org. apache. struts2. dispatcher. mapper.
DefaultActionMapper
<! --指定 Struts 2 使用的 FreeMarker 管理器 -->
struts. freemarker. manager. classname = org. apache. struts2.
views. freemarker. FreemarkerManager

3.2 间接配置属性示例
间接配置就是在 struts. xml 文件中配置,对于 3.1
中的示例,配置方法如下:
<constant name = " struts. configuration" value = " org. apache.
Struts2. config. DefaultConfiguration" />
<constant name = " struts. enable. DynamicMethodInvocation"
value = " true" />
<constant name = " struts. i18n. reload" value = " true" />
<constant name = " struts. locale" value = " zh_CN" />
<constant name = " struts. i18n. encoding" value = " GBK" />
```



```
<constant name="struts.objectFactory.spring.autoWire" value="name" />
<constant name="struts.enable.SlashesInActionNames" value="true" />
<constant name="struts.serve.static.browserCache" value="false" />
<constant name="struts.configuration.files" value="struts-student.xml, struts-teachet.xml" />
<constant name="struts.mapper.class" value="org.apache.struts2.dispatcher.mapper.DefaultActionMapper" />
<constant name="struts.freemarker.manager.classname" value="org.apache.struts2.views.freemarker.FreemarkerManager" />
```

4 结束语

Struts2 是强大的 Java Web 开源框架,是基于 POJO 的 Action 的 MVC Web 框架。Struts2 的优点主要体现在^[12]:基于 MVC 架构,框架结构清晰,开发流程一目了然,开发人员可以很好地掌控开发的过程;使用 OGNL 进行参数传递,可以方便地获取 Request、Attribute、Application、Session、Parameters 中的数据;强大的拦截器,Struts2 的拦截器是一个 Action 级别的 AOP,许多特性都是通过拦截器来实现;易于测试,Struts2 的 Action 都是简单的 POJO,可以方便地编写测试用例;易于扩展的插件机制,只需要将所需要的 Jar 包放到 WEB-INF/lib 文件夹中,在 struts.xml 中作一些简单的设置就可以实现扩展;模块化,可以通过将配置信息拆分成多个文件、把自包含的应用模块创建为插件、将与特定应用无关的新功能组织成插件等方法来将应用程序模块化;全局结果与声明式异常,为应用程序添加全局的 Result,和在配置文件中对异常进行处理。所有这些优点的体现,依赖于强大的配置文件。

文中的研究内容,降低了对软件开发人员的技术要求,提高了软件开发劳动生产率,改变了传统的软件

开发模式^[13]。是 Struts2 的核心技术,有利于团队成员并行工作,对从事 Struts2 框架的开发人员具有很高的参考价值。

参考文献:

- [1] 宋士安,邹俊伟,刘丽华. 基于 Struts+Spring+Hibernate 缺陷管理系统实现[J]. 计算机技术与发展,2012,22(2):146-148.
 - [2] 李 纲. Struts2 权威指南[M]. 北京:电子工业出版社,2008.
 - [3] Liu Y J, Kang J C, Lu W F. Overview of Model-driven Architecture[J]. Computer Science,2006,33(3):226-228.
 - [4] 吕 凯. 基于 MVC 设计模式的 Struts 框架的应用研究[J]. 沈阳工程学院学报(自然科学版),2010,6(4):366-368.
 - [5] Brown D, Davis C M. Struts2 in Action[M]. America: Manning Publications Co,2008.
 - [6] 郑若颖. 一种对 Struts 配置文件加载过程的改进方法[J]. 大众科技,2007,9(9):89-90.
 - [7] 白广元. Java Web 整合开发完全自学手册[M]. 北京:机械工业出版社,2009.
 - [8] 张 毅. 基于 Struts 框架的 J2EE WEB 应用研究与实现[D]. 成都:西南交通大学,2006.
 - [9] Roughley I. Practical Apache Struts2 Web 2.0 Projects[M]. [s.l.]: Springer,2007.
 - [10] 欧阳宏基,马广平,葛 萌. 基于 Struts 框架的 Web 应用研究与实现[J]. 计算机与数字工程,2010,38(3):197-200.
 - [11] 迟却懂. struts.properties 配置文件[EB/OL]. 2011-06-08. <http://czqjay4818.blog.163.com/blog/static/618408820093943021934/>.
 - [12] 阿 锋. struts2 优点[EB/OL]. 2011-06-08. <http://czqjay4818.blog.163.com/blog/static/618408820093943021934/>.
 - [13] Ren Y C. Research and Application on Automatic Generation Technology of JavaScript Input Validation[J]. Advances in Intelligent and Soft Computing,2012,148(1):595-600.
- +++++
- (上接第 76 页)
- ing:Bringing order to the web[M]. Palo Alto,California:Stanford University Press,1998.
- [5] 李稚楹,杨 武,谢治军. PageRank 算法研究综述[J]. 计算机科学,2011(S1):185-188.
 - [6] 李 凯,赫枫龄,左万利. PageRank-Pro-一种改进的网页排序算法[J]. 吉林大学学报(理学版),2003(2):175-179.
 - [7] Brin S,Page L. The anatomy of a large scale hypertextual web search engine[C]//Proceedings of the Seventh International World Wide Web Conference. Brisbane:ACM Press,1998:107-117.
 - [8] Ling Zhang,Zheng Qin. The Improved Pagerank in Web Crawling[C]//The 1st International Conference on Information Science and Engineering ICISE. Piscataway, N. J.: IEEE Press,2009:1889-1891.
 - [9] 邵晶晶,李 波,刘汉平. PageRank 的改进算法-调整阻尼因子[J]. 应用数学,2008(S1):57-61.
 - [10] 高宝军. Web 结构挖掘中 PageRank 算法优化研究[D]. 兰州:兰州大学,2011.
 - [11] 李远方,邓世昆,闻玉彪,等. Hadoop-MapReduce 下的 PageRank 矩阵分块算法[J]. 计算机技术与发展,2011,21(8):6-9.
 - [12] Fu H H, Dennis K J L, Tsai H T. Damping factor in Googlepage ranking[J]. Applied Stochastic Models Business and Industry,2006,22(5-6):431-444.

Struts2框架核心配置文件的研究与应用



作者：[刘艳春， 洪晓慧](#)
作者单位：[刘艳春\(渤海大学 大学计算机教研部, 辽宁 锦州121013\)](#)，[洪晓慧\(中国联通公司锦州分公司, 辽宁 锦州121000\)](#)
刊名：[计算机技术与发展](#)
英文刊名：[Computer Technology and Development](#)
年，卷(期)：2013(2)

本文链接：http://d.g.wanfangdata.com.cn/Periodical_wjtz201302021.aspx